Critic user's guide

Author: Alberto Otero de la Roza, Miguel Álvarez Blanco, Ángel Martín Pendás, Víctor Luaña.
Contact: alberto@carbono.quimica.uniovi.es
Address: Departamento de Química Física y Analítica, Universidad de Oviedo, 33007 Oviedo, Spain.
Version: 1.0

Contents

Introduction

Critic is a full-fledged program for the topological analysis of solid state electron densities. There are three main objectives in critic:

- Provide a generalized platform to interface with as many solid-state programs as possible. As of this version, critic understands the densities provided by WIEN2k (FPLAPW method), and PI (aiPI method). Critic encapsulates common methods for solid state topological analysis and applies them, independently of the source of the electron density.
- Allow the implementation of new algorithms for the integration of properties in atomic basins. Only the basic bisection (PROMEGA) algorithm, by T. Keith, is operative now.
- Extend the range of atomic and molecular properties calculated via the QTAM partition of space. Right now, only the atomic volumes and atomic charges are calculated in all versions. Stress and force are also obtained in the PI version of the code.

Quantum Theory of Atoms in Molecules (QTAM)

In the Quantum Theory of Atoms in Molecules (QTAM), the space is partitioned in disjoint regions, called atomic basins, each corresponding to an atom in the molecule. This partition is not arbitrary, as it is the only one that allows a unique local definition of the momentum based operators, including the kinetic energy. Consequently, rigorous average values for quantum mechanical observables can be defined, including atomic charges, volumes, etc. This basins are characterized by its surfaces: the flux lines generated by the gradient of the electron density do not traverse them. Thus, we call the boundary of the atomic basins zero-flux surfaces (ZFS). Note that an atomic basin is therefore composed of all the points for with omega-limit at the corresponding atomic nucleus.

The critical points of the dynamical system associated with the density are also meaningful in QTAM. The general portrait of the electron density is a strongly accumulated function around the nuclei, and exponentially decaying towards the interatomic space.

QTAM considers two nuclei are bonded if they share an interatomic surface (IAS), i. e., a sheet of the atomic basin. Equivalently, they are bonded if there is a first-order saddle point and a pair of gradient lines which connects them. This saddle point is called a bond critical point (bcp).

Reversing the argument, the minima of the electron density, called cage critical points (ccp), are also associated to repulsion basins, the set of points with alpha- limit at the considered ccp. Two ccps are bonded if there is a second order saddle point, called a ring critical point (rcp), and a pair of gradient lines connecting them (equivalently, they share part of the boundary of the repulsion basin). For completeness, and also because maxima of the electron density can appear at non-nuclear positions, we will call the maxima nuclear critical points (ncp). The non-nuclear maxima (nnm), which have received attention in the past years, for topological purposes behave completely as real atoms, so we will not distinguish between them.

Also, we call the IAS of an atomic basin nuclear IAS (bcp-IAS) and the corresponding topological features of a repulsion basin, cage IAS (rcp-IAS). The bcp- and rcp- prefix express the fact that both two-dimensional manifolds respectively contain one and only one bcp or rcp, which is a limit (omega- or alpha- resp.) of the gradient lines which make up both IAS.

Two atomic basins touching generate a bcp and a pair of gradient lines which connect the two nuclei. The union of these topological features is a line with structure ncp-bcp-ncp. This line is called a bond path and the full set of bond paths is the molecular graph. The molecular graph is equivalent to the symbol-and-stick diagrams the chemists have been drawing for centuries. In analogy to bond paths, we define the ring paths: flux lines connecting ccp-rcp-ccp.

Consider now a bcp-IAS. The flux lines springing from a neighbourhood of the bcp exhaustively cover the IAS. These lines end up at another critical point (in periodic infinite structures all atomic basins are finite. However, there is no problem in assuming a critical point at the infinite, so this discussion is valid also for molecular systems). These terminal CPs can be divided in two classes: stable and unstable. These names correspond to the dynamical terminology: a stable CP attracts all the gradient lines in its neighbourhood while an unstable CP repels most of them. Indeed, the manifold made from the points in the bcp-IAS whose alpha-limit is a stable CP is bidimensional, while it is unidimensional for an unstable CP. Intuitively this means that most of the gradient lines starting from the bcp will end up at an stable CP, while one, and only one, gradient line will terminate at an unstable CP.

User's reference

Data input

WIEN2k version

- **CRYSTAL environment**

```
CRYSTAL
  STRUCT file.s
  CLM file.s
```

```
  CLMDOWN file.s
ENDCRYSTAL
```

The external density specification is done in the CRYSTAL environment. The total electron density in WIEN2k is written to a single clmsum file (in non-spinpolarized calculations) or to clmup and clmdn files, each containing the spin contributions to the total density. The crystal environment allows the following keywords:

- **STRUCT file.s**
  Specification of the path to the struct file, containing all the structural information about the crystal, including symmetry operations. The names of non-equivalent atoms must be different from each other, for they are used as the names of the output files for the basin plotting routines. For the same reason, try not to use names that contain blanks.
- **CLM file.s**
  Path to the first density file. In non-spinpolarized cases, the clmsum file is indicated with this keyword. In spinpolarized cases, any of clmup and clmdn can go here.
- **CLMDOWN file.s**
  The other density file in a spinpolarized calculation.

PI version

- **CRYSTAL environment**

```
CRYSTAL
  SPG spg.s
  CELL a.r b.r c.r alpha.r beta.r gamma.r
  NEQ x.r y.r z.r file.s
  ...
ENDCRYSTAL
```

Beginning of the crystal structure specification. The PI version of critic uses the files containing the description of the wave function of each of the non-equivalent in the unit cell. In these files, there is no structural information such as the space group or the atomic positions so this must be provided by the user using the following keywords.

- **SPG spg.s**
  Enter the crystal space group using the following conventions:
  * Input space group symbol using the international notation.
  * Each different symmetry element must be separated by a blanc.
  * Characters in a symmetry element must be given as a single word. This is the case for subindices.
  * Upperscore is represented by - preceding to the element.
  * Letter case is irrelevant.

Examples	
Enter...	as...
$Fm3m$	f m 3 m
$P2_12_12_1$	p 21 21 21
$P4/mmm$	p 4/m m m
$R\bar{3}c$	r -3 c

- **CELL a.r b.r c.r alpha.r beta.r gamma.r**

  Cell parameters in bohr and sexagesimal degrees.

- **NEQ x.r y.r z.r file.s**

  Add an atom to the main cell, at the position given by x.r y.r z.r (cryst. coords.). The external file file.s contains the information needed to build up the group wave function and density for that atom.

Fundamental tasks

- **POINT x.r y.r z.r**

  Requests topological information about an arbitrary point in the unit cell. The crystallographic coordinates of the point are x.r, y.r and z.r. The level of information provided by critic depends on the character of the point. If it is an ncp, only mod(grad rho), rho and lap are provided. In the general case, also the gradient vector, hessian matrix eigenvectors and eigenvalues and projection of the gradient on the minimum eigenvalue eigenvector of the hessian are provided. Also, if the point has -1 signature, ellipticity is calculated.

- **CHECK environment**

```
CHECK
  x.r y.r z.r
  ...
ENDCHECK
```

  The CHECK environment tests the points indicated within (crystallographic coordinates) to determine if they are CPs. If a point is found to be a CP, then CHECK calculates its properties. At the end of the check job, a synopsis of the CPs known to critic is listed, including bcp connectivity and rcp connectivity. Note that a CHECK that exhausts the non-equivalent CPs (perhaps skipping the atoms known in advance) is equivalent to an AUTO run, so FLUXPRINT, INTEGRALS, etc. can be run after it.

- **LINE x0.r y0.r z0.r x1.r y1.r z1.r npts.i**

  Calculate a line from x0 to x1 with npts points. By default, LINE determines the full properties of every point. At the end of the LINE task, a list with crystallographic coordinates, rho and lap values is written also.

- **FLUXPRINT point {-1|+1|0} x0.r y0.r z0.r step.r eps.r**

  Prints a flux line starting from point x0 (cryst. coords.). -1 issues a descending gradient path and +1 an ascending path. 0 is both -1 and +1, and writes a pair

of flux lines. step.r is the maximum step for the leapfrog algorithm. eps.r is the gradient norm threshold used in the stop criterion.

By default, FLUXPRINT prints a list of points in the path, containing the coordinates, rho and the cartesian derivatives of rho up to second order. There is a list for every path calculated. A more detailed description of the FLUXPRINT keyword is given later in this guide.

- **NEWTON x0.r y0.r z0.r**

  Start a CP search at x0 (cryst. coords.), using the Newton-Raphson algorithm. The NR algorithm is much superior in CP finding tasks than amoeba or any other algorithm we have tested to this day.

  Full information about the final CP is printed.

- **AMOEBA keyword**

```
AMOEBA 1 opt.r s1x.r s1y.r s1z.r s2x.r s2y.r s2z.r
         xopt.r yopt.r zopt.r ftol.r [abstol.r]
AMOEBA 2 opt.r s1x.r s1y.r s1z.r s2x.r s2y.r s2z.r
         s3x.r s3y.r s3z.r xopt.r yopt.r zopt.r ftol.r [abstol.r]
AMOEBA 3 opt.r xopt.r yopt.r zopt.r ftol.r [abstol.r]
```

  Start an amoeba (Nelder-Mead algorithm) CP search. This algorithm applies to the norm of the density gradient, and it requires no information about the hessian matrix. However, it is much slower than Newton-Raphson, and also less reliable.

  The first field in the AMOEBA argument list are the dimensions of the search space. AMOEBA 1 starts a monodimensional search, 2 bidimensional and 3 three-dimensional. The search is constrained to a sphere of radius opt.r * min(a,b,c) where a, b and c are the cell lenght parameters. xopt.r, yopt.r and zopt.r denote the origin of the initial simplex. The optional parameter abstol.r is the absolute tolerance for gradient norm convergence while ftol.r is fractional tolerance for the same quantity. Finally, s1, s2 and s3 indicate the directions in which the simplex vertices are located (**not** the vertex coordinates!). In the three-dimensional case, the (001), (010), (100) directions are used.

  Similarly to newton, AMOEBA prints the information about the final cp point, and the number of iterations used.

- **PRECANALYZE [CORNERS] [POINT x.r y.r z.r] [...]**

  Analyze the precision of the electron density and the norm of its gradient by comparing these values at equivalent points in the unit cell. Corners uses the eight vertices defining the unit cell. Point accepts a position in crystallographic coordinates. An arbitrarily long list of these elements can be given.

- **END**

  Terminates the critic run.

Advanced tasks

- **AUTO {NEWTON|AMOEBA} [gradthr.r [abstol.r [ftol.r]]]**

Automatic location of the CPs of the electron density. newton issues a Newton-Raphson method and amoeba a Nelder-Mead simplex method to find the critical points.

Gradthr is the gradient norm threshold for the optimization. It is used as a convergence criterion even if it is not directly employed in the method (amoeba).

Abstol and ftol are absolute and relative tolerances for gradient norm change between two successive points. Both are used in amoeba. Their default values vary with the selected method: gradthr is 1.d-7 for amoeba and 1.d-15 for newton. Ftol is 1.d-15 and abstol is 1.d-15 in amoeba.

At the end of the automatic CP determination process, an exhaustive list of all the CP inside the unit cell is created by applying the symmetry operations of the crystal space group. Also, the bond paths and rcp -> ccp paths are determined by tracing gradient paths starting from every non-equivalent bcp and rcp.

If the graph (set graph) option is enabled, some information about the attraction or repulsion basin separatrices is gathered. Each separatrix is generated by exactly one bcp (attraction basin) or rcp (repulsion basin). Using the two equal sign hessian eigenvectors, the whole separatrix can be traced following the flux that starts near the generating CP. These flux lines end up in CPs which belong to the limits of the separatrix. They may be classified in 'stable CPs' (if small deviations from a flux line leading to that CP end at the same CP) and 'unstable CPs' (small perturbations on the initial conditions lead to a different CP). The stable CPs associate with a 2D section of the separatrix while there is a unique line which links the separatrix generator and each of the unstable CPs. When the graph option is issued, the stable CPs associated with each bcp and rcp are calculated by exploring a set of starting points around the generator. The unstable CPs are assigned to the CP that links consecutive stable CPs of the separatrix.

- **PLHSEARCH keyword**

```
PLHSEARCH [LIMITS xmin.r xmax.r ymin.r ymax.r zmin.r zmax.r]
          [DIVISIONS numx.i numy.i numz.i]
          [AMOEBA] [NEWTON]
          [FTOL ftol.r] [ABSTOL abstol.r] [RADIUS radius.r]
          [GRADTHR gradthr.r] [MAXIT maxit.r]
```

Perform a critical point search in the interior of the parallelepiped determined by {x,y,z}{min,max} (cryst. coords.). The divisions keyword sets the number of points to calculate in each dimension. Amoeba and newton select the optimization method. The fractional convergence is set by ftol and the absolute convergence for the gradient norm is given by abstol. Radius is a quantity related to the size of the amoeba starting simplex. gradthr is the gradient threshold for CP checking. Maxit is the maximum number of iterations allowed. The use of this keyword is discouraged in favour of AUTO. The default values are: 0. and 1. for ever min and max limit, amoeba, 10 divisions for each axis, ftol = 1d-6, abstol = 1d-8, radius = 1d0, gradthr = 1d-6, maxit = 250.

- **INTEGRALS [thetai.r thetaf.r phii.r phif.r] nr.i ntheta.i nphi.i [NCP id.i]**

Integrate the ncp basins and obtain the topological charge and volume values. Thetai, thetaf, phii and phif are the angular limits for the spherical integration in the basin (polar and azimuthal, respectively). In systems with high symmetry, it

may be useful to set these to a convenient value. By default, however, they are assumed to be 0, pi, 0 and 2*pi respectively. Nr, ntheta and nphi are the number of radial, theta and phi points for the Gauss-Legendre quadrature. If the NCP keyword is issued, critic reads the identifier for a valid nuclear CP, as written in the autocritic final report. The integration will then be restricted to that ncp.

Note that the surface limit search is computationally expensive. Raising the number of radial points will not affect too much on the integration time but the number of angular points calculated is critical.

- **SPHEREINTEGRAL cpid.i rad.r thetai.r thetaf.r phii.r phif.r nr.i ntheta.i nphi.i**

  Integrate in the interior of a sphere centered on the critical point cpid.i (complete cp list) with radius rad.r. The angular limits are given by thetai.r and thetaf.r (polar) and phii.r and phif.r (azimuthal). The Gauss-Legendre quadrature contains nr.i radial points and ntheta.i and nphi.i angular points.

Graphics

- **CELLPLOT**

  Writes a cell.vect file (read by geomview, for example) containing the description of the main cell (cartesian coordinates).

Field representations

- **DENSIPLOT environment**

```
DENSIPLOTBEG root.s [LOG|ATAN|BADER]
 titx.s
 tity.s
 rxa.r rya.r rza.r rxb.r ryb.r rzb.r rxc.r ryc.r rzc.r
 nptsu.i nptsv.i
 niso.i
DENSIPLOTEND
```

  Creates a representation of the charge density in a given plane. root.s is the root of the .iso (contour data for gnuplot), .gnu (gnuplot script file) and .grd (density values at the points of the requested grid) files. A logarithmic representation can be obtained if the LOG keyword is given. The function f(x) = 2/pi*atan(x) is used if ATAN is given. The BADER keyword uses a fixed set of iso-values given by {1,2,4,8}x10^{-3,-2,-1,0,1}.

  The data in the environment specify the characteristics of the plot. titx.s and tity.s are the labels for each axis. Rxa.r, rya.r, rza.r are the crystallographic coordinates of the plane origin. The coordinates of the x-end and y-end of the plot are rxb.r, ryb.r, rzb.r and rxc.r, ryc.r, rzc.r respectively. Niso.i sets the number of isodensity lines in the contour representation.

  The gnuplot script file (root.gnu) is bare-bones and may need further processing to meet the user's needs.

- **DEL2PLOT environment**

```
DEL2PLOTBEG root.s [LOG|ATAN|BADER]
 titx.s
 tity.s
 rxa.r rya.r rza.r rxb.r ryb.r rzb.r rxc.r ryc.r rzc.r
 nptsu.i nptsv.i
 niso.i
DEL2PLOTEND
```

Analogously to densiplot, del2plot creates representations of the laplacian of the electron density in a given plane. The only difference is the presence of a <root>.neg.iso, containing the contour information for the negative values of the laplacian.

- **DOPLOT keyword**

```
DOPLOT [RHO|GRADMOD|{LAPLACIAN|LAP}|GRADIENT]
       [FORMAT {DEFAULT|GNUPLOT|D2D|D3D|CARTESIAN|3DCUBE|CUBE}]
       [FILE root.s] {1D r0x.r r0y.r r0z.r r1x.r r1y.r r1z.r
       npts.i| 2D r0x.r r0y.r r0z.r r1x.r r1y.r r1z.r r2x.r
       r2y.r r2z.r npts1.i npts2.i| 3DC r0x.r r0y.r r0z.r r1x.r
       r1y.r r1z.r npts1.i npts2.i npts3.i}
```

Plot several properties of the system in one, two and three dimensional spaces. The first field stands for the quantity to be represented: density or rho for the electron density, gradmod for the modulus of the gradient of the electron density, laplacian or lap for the laplacian of the electron density, gradient for the gradient of the electron density.

The format keyword modifies the output format. The available formats depend on the dimensions of the plot:

  - 1D:
    - default or gnuplot: a text file containing all the above quantities and a line parameter. Each record in the file represents a point on the requested line.
    - d2d: an input file for d2d, containing only the required property.
    - d3d: not used (same as default).
    - cart: same as gnuplot, but with extended information. Namely, the cartesian coordinates of each point and the components of the hessian matrix are represented.
    - cube, 3dcube: not used (same as default).
  - 2D:
    - default or gnuplot: a text file consisting of 3 fields: the values of u, v and the required property.
    - d2d: not used (same as default).
    - d3d: an input file for d3d.
    - cart: same as gnuplot, but with extended information. Namely, the cartesian coordinates of each point, density, laplacian, gradient modulus and kinetic energy density (not used in WIEN2k version).
    - cube, 3dcube: not used (same as default).

- 3D:
  - default: 3dcube v1.0 format (?)
  - gnuplot, d2d, d3d, cart: not used (same as default).
  - cube: gaussian cube format
  - 3dcube: 3dcube file format (?)

The root for the files generated by DOPLOT can be changed using the file keyword. Next, the number of dimensions of the plot must be chosen with 1d, 2d or 3dc. In a one-dimensional plot, r0x.r, r0y.r, r0z.r, r1x.r, r1y.r and r1z.r are the crystallographic coordinates of the points defining the line to be plotted and npts.i, the number of points on it. In 2d, the origin, end of the x-axis and end of the y-axes are given, followed by the number of points on the x-axis and on the y-axis. Finally, in the 3dc case, two opposite vertices of the parallelepiped are read, followed by the number of points in each dimension.

Flux representations

- **GRDVEC environment**

```
GRDVEC
   FILES rootname.s
   PLANE x0.r y0.r z0.r x1.r y1.r z1.r x2.r y2.r z2.r
   SCALE sx.r sy.r
   PARAM rad1.r rad2.r rad3.r step.r endpt.r proj.i
   ORIG x.r y.r z.r atr.i up.i down.i
   CP id.i up.i down.i
   CPALL
   BCPALL up.i down.i
   RBCPALL bup.i bdown.i rup.i rdown.i
   CHECK
        x.r y.r z.r
        ...
   ENDCHECK
   {RHO|DENSITY|LAP|LAPLACIAN} [LOG|ATAN|BADER] [nptsu.i nptsv.i niso.i]
ENDGRDVEC
```

Plots a plane containing the gradient paths originating from a set of points. The GRDVEC environment accepts a set of lines (in any order) specifying the type of plot.

With FILES the user sets the root name of the output files containing the information for the plot (default: 'grdvec'). These files include:

- rootname-grd.dat : file with gradient path data.
- rootname-rho.dat : file contaning values of the electron density on a grid, if DENSITY or RHO are given.
- rootname-lap.dat : file contaning values of the laplacian of the electron density on a grid, if LAPLACIAN or LAP are given.
- rootname-iso.dat : contour lines for the electron density.
- rootname-isopos.dat : contour lines for the positive region of the laplacian.

- rootname-isoneg.dat : contour lines for the negative region of the laplacian.
- rootname.gnu : gnuplot script file generating the merged gradient/contour plot.
- rootname-label.gnu : gnuplot script file loaded in rootname.gnu containing the information for the position of the CPs in the plot plane.

PLANE sets the plane for the plot: x0 is the origin, x1 the end of the x-axis and x2 the end of the y-axis. This plane may contain regions which are traversed by gradient lines originated at critical points located out of the plot plane. If this is the case, the SCALE option allows the user to extend the plane when considering which origins to be included. The sx.r and sy.r are scale parameters. The x-axis extends (sx.r-1)*l(x) in each direction, where l(x) is the axis length. The sy.r variable works the same way. Consequently, the plane determined by the vectors given in PLANE acts as a clipping plane while the scaled plane determines the gradient path origins. The PARAM keyword sets the leapfrog algorithm parameters: rad1.r is the separation for flux origins associated to a nuclear or cage CP; rad2.r is the same for bcp and rcp when sampling the associated 2d surface and rad3.r, when sampling the associated 1d paths. step.r is the integrator step (cartesian coordinates), endpt.r is unused. The proj.i field is either 1 or 0, representing that the flux lines diverging from the plane are to be projected on it or not, respectively. If the PARAM line does not appear, the default values are:

- Rad1.r, rad2.r and rad3.r assume the value of the change variable, set by the CHANGE or SET CHANGE order (see below). If none of these keywords are given, the default value is 1d-2.
- Step.r adopts the value of grdstep, which is given by a GRDSTEP or SET GRDSTEP order (see below). If none of these keywords are given, the default value is 0.1.
- By default, the gradient lines are projected onto the plane (proj.i equals 1).

The ORIG keyword adds an origin of gradient lines to the plot. Its crystallographic coordinates are x, y and z. atr.i is 1 if the point is to be treated as a ncp or ccp (the up and down trajectories start from points located on a sphere centered on the origin) and it is 0 if the point is to be treated as a bcp or ccp (a circle is built around the CP in the plane determined by two eigenvectors whose eigenvalues have equal sign. The remaining eigenvector determines a unique direction.). up.i and down.i are the number of gradient paths to be started in the upwards and downwards direction respectively.

The CP keyword accepts a critical point identifier from the FINAL REPORT found in the output of AUTO. Also, the number of gradient paths in the upwards and downwards directions must be given. A special case is the CPALL keyword, which adds as origins every critical point in the CP list which lays on the selected plane. The number of gradient paths is 36 down for ncps and 36 up for ccps, and 2 up and 2 down for bcps and rcps. The BCPALL keyword is similar to CPALL, except that only the bond critical points are included as origins. If BCPALL is used, the user must supply the number of gradient lines in the upwards and downwards directions. In a similar way, RBCPALL includes bond and ring critical points, and the user must give the number of upwards and downwards gradient paths for bonds (bup.i, bown.i) and rings (rup.i, rdown.i).

The CHECK environment allows the user to enter the crystallographic coordinates of a CP of the electron density to add it as an origin. If the point given is not a CP or if it lies out of the selected plane, it is ruled out of the origin list. The valid CPs are identified and an adequate number of gradient paths are started according to its character: for a ncp and ccp, 36 upwards or downwards and for a bcp or rcp, 2 upwards and 2 downwards.

The keywords RHO (DENSITY is an alias for RHO) and LAP (LAPLACIAN is the same as LAP) generate a plot which merges the gradient paths calculated in grdvec with a contour plot, in the spirit of DENSIPLOT and DEL2PLOT. This contour map corresponds to the electron density if RHO is used and to the laplacian of the electron density, if LAP is given. Analogously to DENSIPLOT and DEL2PLOT, the contour lines may be calculated using a linear spacing between isocurves on a remapped scalar field. If LOG is given, log(rho) and log(abs(lap)) are used while for ATAN, the function is 2/pi*atan(rhollap). The BADER keyword corresponds to a fixed set of iso-values ({1,2,4,8}x10^{-3,-2,-1,0,1}). The number of points on each axis of the grid and the number of contour lines can be optionally given with nptsu.i, nptsv.i and niso.i. By default, they are 100, 100 and 100.

Note that GRDVEC is able to handle non-ortoghonal axis. If the two plane axis determined in the PARAM keyword are non-orthogonal, the final graph will correctly reflect the actual appearance of the plane by conserving the original angle between the x- and y- axis. This is done by transforming the data to an in-plane cartesian reference system.

Also, note that only 2 gradient lines may be traced from bcps and rcps, either upwards or backwards. Thus, for example, bcpall 2 2 is equivalent to bcpall 100 100.

- **FLUXPRINT keyword**

```
FLUXPRINT POINT {1|-1|0} x.r y.r z.r [step.r epsi.r]
FLUXPRINT NCP id.i ntheta.i nphi.i [step.r epsi.r] [LVEC x.i y.i z.i]
FLUXPRINT BCP id.i 1 [step.r epsi.r] [LVEC x.i y.i z.i]
FLUXPRINT BCP id.i {0|-1} n.i [step.r epsi.r] [LVEC x.i y.i z.i]
              [BRAINDEAD|QUOTIENT]
FLUXPRINT RCP id.i -1 [step.r epsi.r] [LVEC x.i y.i z.i]
FLUXPRINT RCP id.i {0|1} n.i [step.r epsi.r] [LVEC x.i y.i z.i]
              [BRAINDEAD|QUOTIENT]
FLUXPRINT CCP id.i ntheta.i nphi.i [step.r epsi.r] [LVEC x.i y.i z.i]
FLUXPRINT GRAPH igraph.i [step.r epsi.r]
FLUXPRINT GRAPHCP igraph.i cpid.i [step.r epsi.r] [LVEC x.i y.i z.i]
FLUXPRINT OPTIONS STDOUT file-sout.s
FLUXPRINT OPTIONS TESSELOUT file-tout.s
FLUXPRINT OPTIONS SPG spg.i
FLUXPRINT OPTIONS EVERY every.i
FLUXPRINT OPTIONS MPOINTS m.i
FLUXPRINT OPTIONS MSTICKS m.i
FLUXPRINT OPTIONS TITLE title.i
FLUXPRINT OPTIONS SHELLS nsh.i
FLUXPRINT END
```

Print gradient paths and related information. The first field after the FLUXPRINT keyword controls the type of order:

- POINT: build a gradient path starting at point (x.r y.r z.r) in cryst. coordinates. step.r is the step (cartesian coordinates) for the walking algorithm. epsi.r is the gradient norm stop criterion. These two keywords mantain this meaning for every FLUXPRINT keyword. Their default values are 0.1 for step and 1e-7 for epsi. The {1|-1|0} field controls the direction of the path. An ascending gradient path is obtained with 1 while -1 issues a descending path. 0 = -1 + 1 makes FLUXPRINT represent both ascending and descending paths.
- NCP: print gradient paths starting from a (small) sphere centered on the nuclear CP identified by id.i (this identifier comes from the complete cp list, written by autocritic). The number of points is controlled by ntheta.i (number of points sampling the azimuthal angle) and nphi.i (number of points sampling the polar angle). id.i specifies a precise ncp in the main cell up to a lattice traslation. The LVEC optional keyword allows the user to enter a lattice vector to displace the represented ncp gradient paths from their position given in the complete cp list written by autocritic. LVEC maintains its meaning in the rest of the FLUXPRINT input description.
- BCP: print gradient paths starting from the vicinity of a bond CP, identified by id.i. If the gradient path is ascending (1 in the fourth field), the (unique) bond path associated to the bcp is represented. If -1 is given instead, the IAS associated to the bcp is sampled starting from a small circle surrounding the bcp, with n.i points on it. With a 0 value, both tasks are performed.

  The two keywords BRAINDEAD and QUOTIENT set the method employed in generating the starting angular grid. With BRAINDEAD, critic uses a uniform angular grid. Using QUOTIENT, the uniform grid is remapped by x^(l1/l2) where l1 and l2 are the two negative eigenvalues at the bcp. This way, the points get accumulated around the bcp with lowest eigenvalue (highest if absolute value is taken).

  By default, QUOTIENT is used.
- RCP: print gradient paths starting from the neighbourhood of a ring CP. The situation is analogous to that of the bcps.
- CCP: print gradient paths starting from the vicinity of a cage CP. Again, the situation is symmetric to the ncp case.
- GRAPH: represent the complete graph in the unit cell and adjacent shells. This means:
  * All the bond paths for which both ncps and the bcp lay inside the main unit cell.
  * All the ring paths for which both ccps and the rcp lay inside the main unit cell.
  * A collection of sticks, representing connections between CPs in an attraction or repulsion IAS. For each bcp and rcp, sticks connecting the IAS generator and all the stable and unstable CPs associated to it are created. Both the generator and the terminal stable or unstable CP must be inside the main unit cell.

  The critical points situated on the boundary of the main cell are also represented.

  The igraph.i value represents the quantity of information that is to be printed. It is a sum of values, each representing an element to plot:

* 1 : print ring paths and sticks associated to the rcps.
* 2 : print bond paths and sticks associated to the bcps.
* 4 : print sticks associated to stable CPs.
* 8 : print sticks associated to unstable CPs.

The 4 and 8 options are only available if an AUTO task was carried out with the option SET GRAPH.

– GRAPHCP: represents a partial graph in the unit cell. Only the paths and sticks related to the CP cpid.i (complete cp list, traslated by LVEC, if present) is plotted. The behaviour of this keyword varies accoring to the type of CP and the igraph.i value, whose meaning is close to the explained above:

  * ncps: 1, ring paths for which the ncp is a stable CP of its generated IAS; 2, bond paths for the bcps that are directly linked to the ncp; 4, sticks for the stable CPs associated with the bcps bonded to cpid.i; 8, sticks for the unstable CPs associated with the bcps bonded to cpid.i.
  * bcps: 1, bond paths for cpid.i; 4, stable CPs associated to cpid.i; 8, unstable CPs associated to cpid.i.
  * rcps: 2, ring paths for cpid.i; 4, stable CPs associated to cpid.i; 8, unstable CPs associated to cpid.i.
  * ccps: 1, ring paths for the rcps that are directly linked to the ccp; 2, bond paths for which the ccp is a stable CP of its generated IAS; 4, sticks for the stable CPs associated with the rcps bonded to cpid.i; 8, sticks for the unstable CPs associated with the rcps bonded to cpid.i.

  The sticks (information associated to stable and unstable CPs of an IAS) are only represented if an AUTO was performed with the SET GRAPH option enabled.

– OPTIONS: several parameters of FLUXPRINT can be changed using the options keyword. The default output of FLUXPRINT is the standard output, but it can also create a modifiable input file for tessel. As, usually, lots of information will be generated, it may be interesting to redirect it to a separate file. This can be done using the stdout keyword.

  By default, a tessel input file is not written by FLUXPRINT. With the tesselout keyword, this feature is activated, and the name of this file is filetout.s. Note that the tessel input file *requires* the specification of the space group for the system. See above for details on the syntax. Note that this is only required in the WIEN2k version, for the PI version of critic require the space group to be given in the crystal environment.

  Sometimes the tessel input file contains too many points which results in a very slow rendering. The every keyword makes critic write only one in every every.i points for each gradient path.

  The mpoints and msticks keywords control the size of the arrays that contain the path and stick information in FLUXPRINT. Specifically, mpoints must be greater than twice the number of points of the longest gradient path calculated. The default values are 40000 for mpoints and 3000 for msticks. The title for the standard output file can be set using the title keyword. The default title is 'gradient paths'.

Finally, the shells keyword applies only to graph and graphcp. It represents the number of unit cell shells where the graph is going to be plotted. Thus, 0 represents the main unit cell; 1, the main unit cell and its 26 neighbours; and so on. By default, shells adopts the -1 value, which is equivalent to 0 for the graph keyword and means that the partial graph generated in graphcp is not expanded through symmetry.

All the options except the shells option remain fixed once the first printing command is issued (point, ncp, bcp, rcp, ccp, graph or graphcp).

Atomic basin representations

- **TRIANGULATE ntheta.i nphi.i surfile.s**

  Tesselate the unit sphere using a triangulation with ntheta.i and nphi.i angular points. Then use these rays to find the zero-flux surface of each attraction basin (associated to the ncps found by autocritic) using a bisection method. The results are written to a OFF file with name surfile.s.

- **BASINPLOT [lvl.i [delta.r [phasetheta.r [phasephi.r]]]]**

  Plot the attraction basin of each ncp found in autocritic. If lvl.i > 0, BASINPLOT starts with an octahedron and recursively subdivides it lvl.i times. If lvl.i < 0, the starting polyhedron is a cube, which is subdivided -lvl.i times. Using the resulting points, the zero-flux surface is found using a bisection method. delta.r is the precision (cryst. coords.) with which the surface is determined in each ray. phasetheta.r and phasephi.r rotate the original polyhedron. The default values are 4, 1d-3, 0 and 0 respectively.

  Note that if lvl.i > 0, the triangulation generates a geomview OFF file while for lvl.i < 0, the cubication generates BASIN files.

- **PROGRESSIVE [lvl.i [ncycle.i [cutangle.r [delta.r [phasetheta.r [phasephi.r]]]]]**

  Analogous to BASINPLOT, but after the basin is obtained, the low quality faces are refined. The refinement process is done by splitting the faces which form an angle greater than cutangle.r (sexagesimal degrees) with the adjacent faces. This refinement process is repeated ncycle.i times. The new points generated at each refinement step set the need for a re-triangulation of the convex polyhedron which is achieved using the external qhull (cita) program. The default values are 2, 4, 40d0, 1d-3, 0d0, 0d0.

- **BASINDENSITY [lvl.i [npts.i [delta.r [phasetheta.r [phasephi.r]]]]]**

  Sample the electron density in npts.i rays centered at every ncp using a tesselated cube with lvl.i subidivision levels. delta.r is the precision required in the determination of the zero-flux surfaces. phasetheta.r and phasephi.r rotate the original cube. The default values are 3, 11, 1d-3, 0, 0.

  BASINDENSITY writes a DBASIN output file for every ncp known to critic (use autocritic first).

- **BUNDLEPLOT x.r y.r z.r surfile.s [lvl.i [delta.r [phasetheta.r [phasephi.r]]]]**

  Plot a representation of the primary bundle in which interior the point x.r y.r z.r (cryst. coords.) is located. The output file is surfile.s with OFF format. The

process starts by creating a polyhedron around the given point. If lvl.i > 0, an octahedron is used and a cube if lvl.i < 0. The polyhedron is subdivided abs(lvl) times. For each of the rays determined by its vertices and the starting point, determine the surface of the primary bundle using a bisection algorithm. A point is considered to be inside the primary bundle if it shares the omega- and alpha-limits with the original point. Defaults for lvl.i, etc. are: 4, 1d-3, 0d0, 0d0.

Options

- **TITLE title.s**

  Sets the title of the critic run.

- **SET PRINT printlvl.i**

  PRINT [printlvl.i]

  Sets the printing level. The default value is 0, and increasing values give more information on output. For printlvl = 1, the additional output consists of:

  - An environment anaylisis of the non-equivalent CPs found in check and auto is carried out.
  - Integrals gives the progress of the integration process by writing the number of rays for which the zero-flux surface limit has been located.
  - BASINPLOT and TRIANGULATE write a header containing some useful information about the plot.

  Additionally, for printlvl = 2, the environment analysis in auto and check is even more verbose. If only PRINT or SET PRINT is given, use printlvl.i = 1.

- **SET DEBUG [debuglvl.i]**

  DEBUG [debuglvl.i]

  Sets the debug level. The default value is 0, for which time records of the main program process and setvariables are kept. If debuglvl.i is greater than 0, the time spent in each process is calculated and written at the end of the critic run. Note that this feature will make critic a little slower: some 2-3% of the total wall time. If only DEBUG or SET DEBUG is given, use debuglvl.i = 1.

- **SET CHANGE change.r**

  CHANGE change.r

  Sets the displacement (in bohr) from a near cp to start the integration of a gradient path. It is used when building bond and ring paths, IAS associated to bcps and rcps and spheres around a ncp or a ccp. The default value is 1d-2.

- **SET GRDSTEP step.r**

  GRDSTEP step.r

  Step for the gradient path integration. This value is applied in every flux traced, except for FLUXPRINT, where a step is required as input. Specifically, this includes the bond path and ring path determination in AUTO and CHECK, the gradient path tracing in GRDVEC and the paths in the bisection algorithm for the localization of the zero-flux surface. The default value is 0.1.

- **SET IWS [depth.i]**

  IWS [depth.i]

  In the automatic localization of CPs, start from the irreducible wedge of the Wigner-Seitz cell (IWS) applying depth.i subdivisions on edges, faces and the interior of the polyhedron. The resulting points are used as seeds for the Newton-Raphson algorithm. The default depth is 2.

- **SET NOIWS [depth.i]**

  NOIWS [depth.i]

  Same as IWS but starting from the whole unit cell. The default depth is 2.

- **SET GRAPH**

  GRAPH

  In AUTO, calculate the stable CPs associated to each bidimensional manifold generated by bond and ring critical points. Then, determine the unstable CPs by examining the connectivity. Note that the complete set of stable CPs of the IAS is required and this is not always available by simple gradient path exploration. Consequently, this option may fail to give all the stable and unstable CPs.

- **SET COREDENS coredens.r**

  COREDENS coredens.r

  A point for which the electron density is greater than coredens.r can be considered to belong clearly to a certain nucleus in its neighbourhood. This value finds use only in the leap-frog flux integrators. For a gradient path heading upwards, critic checks the proximity to a nucleus only if the density value is greater than coredens.r. Default: 0.2d0

- **SET CUTDENS cutdens.r**

  CUTDENS cutdens.r

  Cutoff for the electron density. In the pi version, the electron density at a point in the main cell is generated as a sum over atomic centers. At the beginning of the run, the identity of the centers contributing to rho in the main cell is determined. This is done by calculating the radius for which the atomic (in vacuo) electron density has a value lower than cutdens.r. The distance for which this happens is then compared to the distance to the main cell. Sucessive shells of unit cells are thus built up until no more atoms contribute more than cutdens.r to the density in the main cell.

  In the WIEN2k version, CUTDENS and INNERRAD keywords have no effect.

  Default: 1d-12.

- **SET INNERRAD innerrad.r**

  INNERRAD innerrad.r

  Minimum distance (in bohr) for which atomic centers participate in the electron density buildup.

  Not used in WIEN2k version (see cutdens).

  Default: 1d0.

- **SET EXACT [.true.|.false.]**

  EXACT [.true.|.false.]

  EXACT requires that no approximation is made to the electron density or its gradient.

  EXACT is used only in the PI version.

- **SET APPROXIMATE [.true.|.false.]**

  APPROXIMATE [.true.|.false.]

  SET APROXIMATE [.true.|.false.]

  APROXIMATE [.true.|.false.]

  When APPROXIMATE is given, critic builds a table with values of the contribution of each atom to the electron density and its gradient at the points of a radial mesh. Every value of rho and grad is then calculated by summing the contributions of each ion, which are determined by polynomial interpolation on the mesh. Note that when APPROXIMATE is set, the hessian matrix is *not* calculated.

  APPROXIMATE is used only in the PI version.

- **SET NEWGNUPLOT**

  NEWGNUPLOT

  Gnuplot input files are written for newer (>=4.2.0) versions of gnuplot.

- **SET OLDGNUPLOT**

  OLDGNUPLOT

  Gnuplot input files are written for older (<4.2.0) versions of gnuplot.

- **SET INTERACTIVE**

  INTERACTIVE

  Deprecated. --

- **SET NONINTERACTIVE**

  NONINTERACTIVE

  Deprecated. --

- **SET FILTERFACTOR**

  FILTERFACTOR

  Deprecated. --

- **SET RADIUS**

  RADIUS

  Deprecated. --

Tools

Off2off

Off2off is a program to transform and combine OFF and COFF three-dimensional object description files. To run the program, use:

```
off2off [OPTIONS] input_file(s)
```

where input_files are one or more files written in OFF or COFF format. The output, also written in OFF or COFF format is written to the standard output unit, as a default.

The options for off2off are:

- -c : give (solid) color to OFF files (default).
- -u : remove color from COFF files.
- -z : translate coordinates origin to the geometric center of files.
- -m : translate coordinates to the mass center of input points.
- -s : scale point (x,y,z) coordinates to [-1,1].
- -b : read and write BASIN files instead of the standard OFF.
- -d : read and write DBASIN files.
- -o output : write results to given file instead to stdout.
- -t matrix : read in a group of transformation matrices from the given file.

Basin2off

Basin2off uses the data contained in one or more BASIN data files and prepares OFF and COFF files to plot colorpam renderings of a scalar property on the atomic basins surface. It is run:

```
basin2off [-c] [-g] [-l] [-a] [-L] [-A] [-n num] input_files....
```

The list of input_files will be analyzed together to produce a common color scale. For each input file a COFF or OFF file will be produced, its name being that of the input with the suffix '.basin' stripped and substituted by '.coff' or '.off'. The list of options is:

- -c : used together with -n 0 will produce colored OFF files. Each basin input file will be given a different solid color.
- -g : use a simple scale based on gray values, instead of the default color scale.
- -l : assign the colors according to a logarithmic scale rather than the default linear scale.

- -a : use a atan() mapping function instead.
- -L : experimental: atan(log()) map
- -A : experimental: atan(atan()) map
- -n num : the scalar property used to produce the COFF files will be the num-th one contained in the BASIN files. (Default: num = 1). (Special case: num=0 will produce OFF files with no color defined for the basin faces).

Basinmerge

Basinmerge is a script to copy by symmetry and combine the BASIN/DBASIN/OFF/-COFF datafiles of several atoms in the crystal. It uses the tessel, off2off and basin2off programs. The program interprets a collection of orders/instructions written according to the next language:

- **FILETYPE [OFF|COFF|BASIN|DBASIN]**

  Type of file to be processed.

- **NEQION name.s file.s x.r y.r z.r**

  Enter basin description for each non-equivalent ion

- **CRYS2CART**

```
c11.r  c12.r  c13.r  c14.r
c21.r  c22.r  c23.r  c24.r
c31.r  c32.r  c33.r  c34.r
c41.r  c42.r  c43.r  c44.r
```

  Matrix that transforms from crystal to cartesian coordinates. It is written in the critic output.

- **CART2CRYST**

```
d11.r  d12.r  d13.r  d14.r
d21.r  d22.r  d23.r  d24.r
d31.r  d32.r  d33.r  d34.r
d41.r  d42.r  d43.r  d44.r
```

  Matrix that transforms from cartesian to crystal coordinates. It is written in the critic output.

- **COPY fromname.s toname.s fileto.s**

```
t11.r  t12.r  t13.r  t14.r
t21.r  t22.r  t23.r  t24.r
t31.r  t32.r  t33.r  t34.r
t41.r  t42.r  t43.r  t44.r
```

  Create a copy of the basin file of a NEQ ion into an equivalent position. The transformation matrix that is entered should act on the crystallographic coordinates, and it is produced by critic.

- **MERGETO destfile.s file1.s [file2.s ...]**

  Merge a collection of files into a single one.

- **BASIN2OFF order.s**

  Run directly the basin2off program.

- **OFF2OFF order.s**

  Run directly the off2off program.

Test files

Some comparative tests are provided with critic. These can serve as templates for new calculations and to check the installation. The systems included are: CaTiO3 perovskite structure, Cu3N, gamma-Fe and alpha-Fe. Only the most reprsentative keywords have been chosen for the tests.

Appendix

Related programs and output file formats

Some programs required or just useful when working with critic are:

- **gnuplot**: most of the output critic creates are text files with column organized grid data, readily formatted to use with gnuplot. Also, in some tasks, critic also generates the gnuplot scripts required for the representation.
- **geomview**: some atomic basin plots are created in geomview's OFF and COFF format, which are described below.
- **tessel**: tessel is a generalized plotting program for crystalline structures. Critic provides input files for tessel (mainly in FLUXPRINT). Also, BASIN and DBASIN files, containing the information about the atomic basins are generated, which are read and plotted by tessel.
- **qhull**: from the debian repository, 'Qhull computes convex hulls, Delaunay triangulations, halfspace intersections about a point, Voronoi diagrams, furthest-site Delaunay triangulations, and furthest-site Voronoi diagrams. It runs in 2-d, 3-d, 4-d, and higher dimensions'. It is **required** if the user calls the PROGRESSIVE order, which creates a representation of the atomic basin and smooths the resulting surface (see below).
- **d2d, d3d**: two- and three-dimensional plotting programs.
- **molekel**: molekel understands the gaussian CUBE format, which contains three-dimensional representations of the topological scalar fields calculated in critic.

The description of the output file formats in critic follows.

BASIN files

The surface of an atomic basin is approximated in critic by a polyhedron, with vertices on the rays on which the zero-flux surface has been determined. This surface, named basin from now on, is approximately described as a number of polygonal facets, each being a ordered list of vertices.

Let us consider, in addition, several scalar properties evaluated on the vertices of the surface. We can select any of this properties to create a color map of the scalar property on the basin surface.

This files can only be used to analyze those surfaces that are monovaluated in spherical coordinates (i.e. surfaces that have one and only one value of the radial coordinate for every angular point).

The structure of the BASIN files is:

- **Rec. 0** : "#" comment.

  Comment lines starting with "#" may appear anywhere.

- **Rec. 1** : nvert, nface, nedge

  Number of vertices, faces and edges of the polyhedron.

- **Rec. 2**: npropty

  Number of properties (scalars or scalar components) that will be given for each point.

- **Rec. 3** : (propname(i), i = 1, npropty)

  Alphanumerical label for each property. Blank characters (one or more) separate the properties. Notice that vectors are given as three separate components, etc.

- **Rec. 4i, i = 0, nvert-1** : x(i), y(i), z(i), (prop(j,i),j=1,npropty)

  For each vertex: the cartesian coordinates and the values of all the properties given for this point.

- **Rec. 5j, j = 0, nface-1** : nv, (ivert(k,j), k=1,nv)

  The number of vertices of this face is nv (nv=3 in the triangular tesselations and 4 in the quadrilateral ones). For each vertex in this face, the number of order in the previous vertex list. Remember that vertices are numbered from 0 to nvert-1.

DBASIN files

The DBASIN files contain the description of the basin of a point, and the value of a scalar property inside the basin (e.g. the electron density). Those data are used to plot the surfaces of constant value of the scalar property.

A regular grid of NPOINT points is defined along each ray from the origin (excluded) to the limit of the basin for this ray. The scalar property is computed in these points. Notice that the grid is, in principle, different for each ray.

The structure of a DBASIN file is:

- **Rec. 0** : "#" comment

  Comment lines starting with "#" may appear anywhere.

- **Rec. 1** : nvert, nface, nedge

  Number of vertices, faces and edges of the polyhedron.

- **Rec. 2** : npoint, xnuc, ynuc, znuc, rhonuc

  Number of sampled points along each ray. Cartesian (x,y,z) position of the nucleus. Electron density at the nucleus.

- **Rec. 3i, i = 0, nvert-1** : x(i), y(i), z(i), (rho(j,i),j=1,npoint)

  For each vertex: the cartesian coordinates and the values of the electron density at the grid points.

- **Rec. 4j, j = 0, nface-1**: nv, (ivert(k,j), k=1,nv)

  The number of vertices of this face is nv (nv=3 in the triangular tesselations and 4 in the quadrilateral ones). For each vertex in this face, the number of order in the previous vertex list. Remember that vertices are numbered from 0 to nvert-1.

OFF and COFF files

The OFF/COFF files can be viewed and printed with geomview. The structure is:

- **Rec. 0** : "#" comment.

  Comment lines starting with "#" may appear anywhere.

- **Rec. 1** : file_type

  Either OFF or COFF keyword.

- **Rec. 2** : nvert, nface, nedge

  Number of vertices, faces and edges of the polyhedron.

- **Rec. 3** : (x(i),y(i),z(i), [r(i),g(i),b(i),alpha(i)],i = 0, nvert-1)

  Cartesian coordinates (x,y,z) of the vertices. In the case of COFF files, additional information is given regarding the RGB color associated to the vertex, and the degree of transparency (alpha) of the surface at this position.

- **Rec. 4j, j = 0, nface-1** : nv, (ivert(k,j), k=1,nv)

  The number of vertices of this face is nv (nv=3 in the triangular tesselations and 4 in the quadrilateral ones). For each vertex in this face, the number of order in the previous vertex list. Remember that vertices are numbered from 0 to nvert-1.

D3D files

The D3D files represent the value of a z(x,y) function in points defined by a linear grid on the XY plane. This format is implemented for compatibility with old programs. Other programs that read/write this format are: ISOLIN, NIVEL, etc.

The structure of the D3D files is:

- **Rec. 1** : title1
- **Rec. 2** : title2
- **Rec. 3** : title3
- **Rec. 4** : title4

  Four lines of titles used in other programs.
- **Rec. 5** : xmin, xmax, nx, ymin, ymax, ny

  Definition of a linear grid in the XY plane. The grid on the X axis goes from xmin to xmax and has nx points. Similarly, the grid on the Y axis goes from ymin to ymax with ny points.
- **Rec. 6** : ((z(i,j), i = 1, nx), j = 1, ny)

  Value of the coordinate z on the grid points.

Copyright notice